DoctorantMemory’s Handy Guide

# Generating a Trace

To get a trace, you can use the HelloWorld program included. First, it must be compiled.

# 

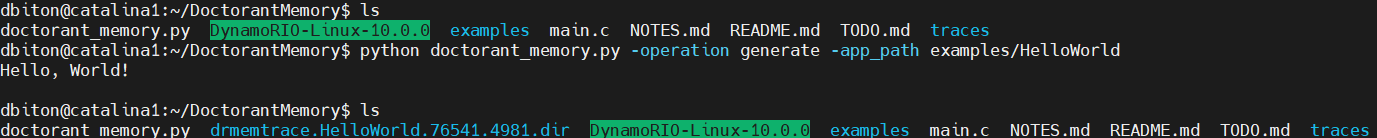
Afterwards, run the following command:

**python doctorant\_memory.py -operation generate -app\_path examples/HelloWorld**

DoctorantMemory creates a file named DOCTORANT\_MEMORY\_TS, with TS being an iso timestamp of the time the trace was created, for example: DOCTORANT\_MEMORY\_20240410T143504Z. This file contains the output of the executable, which is also printed to the console.

This file is automatically deleted, if you would like to keep it, you can use the -keep\_logs flag.

The trace information is written to a folder created by drcachesim in the current working directory.

As you can see below, a trace folder was created:

You can set the path where the folder would be created, with missing folders created if needed:

**python .\doctorant\_memory.py -operation generate -app\_path ./HelloWorld.exe -trace\_path traces**

The folder traces would be created if it does not exist, as well as any more needed folders.

We have included the trace output in the "example trace" folder, in case you are having trouble with generating it.

# Parsing a Trace

There are different trace tools available. The default one simulates a multiple leveled cache and shows statistics about cache hits and misses while running the program.

You can use the following command to run it:

**python doctorant\_memory.py -operation parse -trace\_path drmemtrace.HelloWorld.YOUR\_FOLDER\_NAME.dir/**

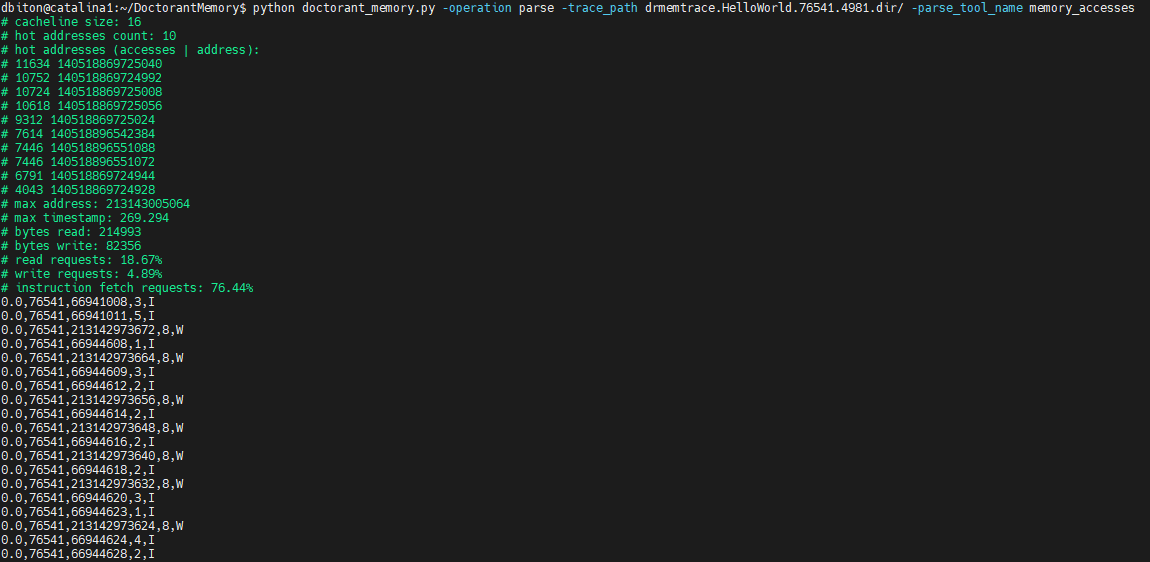
don’t forget to use the name created for your trace in the command.

A screenshot of a computer

Description automatically generated

Another useful tool can present the memory accesses in the format used by you, with an additional header containing useful information.

**python doctorant\_memory.py -operation parse -trace\_path drmemtrace.HelloWorld.YOUR\_FOLDER\_NAME.dir/ -parse\_tool\_name memory\_accesses**



You can use the **-keep\_logs** flag to keep a file containing the console output, as well as a file containing all of the hot addresses sorted in a descending order, named DOCTORANT\_MEMORY\_HOT\_ADDRESSES\_TS with TS being a timestamp.

You can use the **-parse\_ignore\_inst** flag to ignore instruction fetch memory accesses, as well as using the **-parse\_hot\_addresses\_count** and **-parse\_cacheline\_size** to further modify the header data presented.